(12) **EUROPEAN PATENT APPLICATION**

(72) Inventor:
Roy, Sebastien,
c/o NEC Research Institute, Inc.
Princeton, NJ 08540 (US)

(74) Representative:
Patentanwälte Wenzel & Kalkoff
Grubesallee 26
22143 Hamburg (DE)

(54) **Maximum flow method for stereo correspondence**

(57) A method for solving the stereo correspondence problem transforms the problem into a maximum-flow problem. Once solved, the minimum-cut associated to the maximum-flow results in a disparity surface for the entire image at once. This global approach to stereo analysis provides a more accurate and coherent depth map than the traditional line-by-line stereo method. Moreover, the optimality of the depth surface is guaranteed and can be shown to be a generalization of the dynamic programming approach that is widely used in standard stereo. Results show improved depth estimation as well as better handling of depth discontinuities. While the worst case running time is $O(n^3 d^3)$, the observed average running time is $O(n\, d^{1.4})$ for an image size of n pixels and depth resolution d.
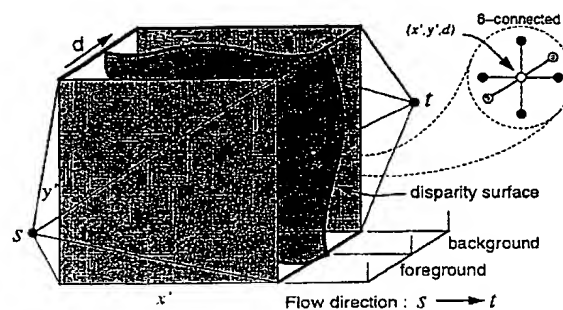
Figure 3

EP 0 871 144 A2

Description

BACKGROUND OF THE INVENTION

5   1. Field of invention:

The present invention relates generally to the field of machine vision and in particular to a method for providing stereo correspondence between two or more arbitrary views of a scene.

10   2. Description of the related art:

It is well known in the art that depth related displacements in stereo pairs always occur along lines, the epipolar lines. Traditionally, this constraint has been most widely used to decompose a matching process into a number of independent matching of pairs of lines. However, the solutions obtained on consecutive epipolar lines can vary significantly
15   and create artifacts (an erroneous match) across epipolar lines that affect in particular object boundaries that are perpendicular to the epipolar lines (i,e. a vertical boundary with horizontal epipolar lines). Consequently, several attempts have been made to relate the solutions of consecutive lines.

In an article by Y. Ohta and T. Kanade, entitled "Stereo by intra and inter-scanline using dynamic programming", IEEE Trans. Pattern Analysis and Machine Intelligence, 7(2): 139, 1996, dynamic programming is used to first match
20   epipolar lines and then iteratively improve the solutions obtained by using vertical edges as reference. In an article by I. J. Cox, S. Hingorani, B.M. Maggs and S.B. Rao, entitled "A maximum likelihood stereo algorithm", Computer Vision and Image Understanding, 63(3):542-567, 1996, a probabilistic approach is used to relate the individual matchings obtained by dynamic programming to improve the depth map quality. Finally, in an article by P.N. Belhumeur entitled "A bayesian approach to binocular stereopsis". Int. J. Computer vision, 19(3):237-260, 1996, a Bayesian approach is
25   described. Unfortunately, the optimality and simplicity of dynamic programming applied to a single pair of epipolar lines cannot be generalized directly to the matching of all epipolar lines at once. For such a case, only iterative or approximate solutions are available. One such iterative or approximate solution is described in Ohta et al, supra.

The dynamic programming approach for finding a matching path between two lines cannot be generalized to matching two full images at once. However, the present inventor has discovered that a formulation using maximum-flow
30   estimation can be used instead which allows matching full images without concern for epipolar lines.

SUMMARY OF THE INVENTION

The above problems are solved and an advance is made over the prior art in accordance to the teachings of the
35   present invention in which a new method is used to perform stereoscopic correspondence between two or more arbitrary views of a scene. The method advantageously solves the stereo correspondence problem which is formulated as a maximum flow in a graph, thereby alleviating many of the problems associated with standard line-by-line stereo methods.

The present invention teaches a method of obtaining stereo correspondence between multiple views of a scene
40   from a sequence of images thereof.

The method is characterized by the steps of:

creating a graph $G = (V, E)$ of the entire scene comprising a set of vertices V, a set of edges E linking the vertices, a source s, a sink t and an edge capacity function $c(u, v)$ representing matching cost where u and v are vertices
45   other than a source or a sink, wherein said set of vertices V comprises said source s, said sink t and a three-dimensional six-connected mesh V^* associated with an x'-, a y'- and a d-axis, said x' and y' being image coordinates and d denoting a disparity and wherein said source s is connected to the front plane of said three-dimensional mesh as viewed in the direction of the d-axis and said sink t is connected to the back plane of said three-dimensional mesh as viewed in the direction of the d-axis; finding a maximum-flow in graph G; extracting a minimum-cut from said
50   maximum-flow; and computing a disparity surface from said minimum-cut.

Stereo matching of images is a process of matching points in the different images of the same scene in order to determine the depth of the points, so called depth map or disparity map, as the output of the stereo matching process. Previous stereo matching processes were limited to arrangements where the cameras recording the image had to be
55   on the same line, i.e., share a common baseline. In addition, prior stereo correspondence was performed on a line-by-line basis using epipolar lines.

The present invention matches points from two or more images when the images are arbitrary views. That is, the positions of the cameras recording the image are not collinear.

A primary advantage of the invention is the ability to match the entire image at one time rather than matching in a line-to-line manner.

The invention relies upon a maximum flow-minimum cut method to perform the stereoscopic correspondence between two or more arbitrary views of a scene. The result is a more efficient, faster and higher quality solution to the stereo correspondence problem.

The invention will be more clearly understood when the following description is read in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram showing epipolar matching;
Figure 2 is a diagram depicting matching of whole images;
Figure 3 is a diagram depicting image matching as a maximum flow problem according to the teachings of the present invention;
Figure 4 is a graph showing the performance of the method of the present method as a function of image size in pixels;
Figure 5 is a graph showing the performance of the method of the present invention as a function of depth for fixed image size;
Figure 6 is a random dot stereogram;
Figures 7A and 7B are disparity maps for the random dot stereogram of Figure 6;
Figure 8 is a stereo pair for a representative image;
Figure 9 is a set of disparity maps for the image of Figure 8;
Figures 10A, 10B and 10C are a set of disparity maps for the image of Figure 8 at three occlusion costs;
Figure 11 is a schematic diagram showing the general stereo framework for a situation having three cameras;
Figures 12A and 12B are a schematic diagram showing a horizontal binocular stereo framework having two cameras; and
Figure 13 is a flow diagram which depicts the overall method of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The standard stereo geometry is composed of two cameras viewing the same scene from different viewpoints. The geometry equations can be derived describing arbitrary multi-image stereo geometry. In the simple case of two cameras separated horizontally by a baseline distance b, the set of corresponding images point $v(x_0, y_0, d)$ associated with image point (x_0, y_0) at disparity d is

$$v(x_0, y_0, d) = \{l_1(x_0, y_0), l_2(x_0 + bd, y_0),\} \tag{1}$$

as derived from Equation 3 below. The usual matching cost function derived from $v(x_0, y_0, d)$ is the sum of square difference of pixel intensities

$$cost(x_0, y_0, d) = \{l_1(x_0, y_0) - l_2(x_0 + bd, y_0)\}^2 \tag{2}$$

Even though the remainder of the discussion uses this stereo framework for simplicity, the invention generalizes directly to consider stereo in the context of multiple images taken under arbitrary camera geometries (i.e., multiple baselines). This naturally extends the traditional two-images, single-baseline framework for stereo.

The general framework of the invention is illustrated in Figure 11. Three cameras C_1, C_2 and C_3 provide the images l_1, l_2 and l_3 for corresponding matching. The reference camera C_0 is the view for which it is desired to compute the disparity map (or equivalently depth map) for every image point. The reference camera may have no associated image. It will rather be reconstructed after stereo matching. As illustrated in Figures 12A and 12B, the standard horizontal binocular stereo setup is easily expressed in this general framework. In that case, the reference view C_0 coincides exactly with the first camera C_1 (Figure 12A). Placing the reference view equidistant between camera C_1 and camera C_2 gives rise to the cyclopean stereo setup (Figure 12B).

A set of n camera $C_1, ..., C_n$ provides n images $l_1, ..., l_n$. A point P_w expressed in the world coordinate system with homogeneous coordinates

$$P_w = [x_w \ y_w \ z_w \ 1]^T$$

can be transformed to the homogeneous point P_i in the coordinate system of camera i by the relation

$$P_i = W_i P_w$$

where

$$W_i = \begin{bmatrix} R_i & T_i \\ 0^T & 1 \end{bmatrix}$$

and R_i is a rotation matrix, T_i is a translation vector and O^T is a null vector. Assuming a pinhole camera model, a point P_i is projected onto the image plane in the projective point P_i by the relation

$$P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = J P_i$$

where J is the projection matrix defined as

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

From a transformed and projected point p_i, the corresponding image coordinates $\dot{p}_i$ are obtained as

$$\dot{p}_i = H(p_i)$$

where H is an homogenizing function

$$H\left(\begin{bmatrix} x \\ y \\ h \end{bmatrix}\right) = \begin{bmatrix} x/h \\ y/h \end{bmatrix}$$

During the process of stereo matching, each image point of I_0 is attributed to a depth z or equivalently a disparity d (defined as $z = 1/d$) and can be expressed as

$$P_0 = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \dot{x}_0/z \\ \dot{y}_0/z \\ 1 \\ d \end{bmatrix} = \begin{bmatrix} \ddot{x}_0 \\ \ddot{y}_0 \\ 1 \\ d \end{bmatrix}$$

in the reference coordinate system C_0. While these three formulations are equivalent, using the disparity allows expression naturally of points that reach an infinite depth. Thus, disparity is used instead of depth.

From this point P_0 it is possible to project back to any camera image $\dot{p}_i$ using the previously defined equations as

$$\dot{p}_i = H(p_i)$$
$$= H(Jp_i)$$
$$= H(JW_i P_w)$$
$$= H(JW_i W_0^{-1} P_0)$$

$$\begin{bmatrix} x_i' \\ \cdot \\ y_i' \end{bmatrix} = H(JW_i W_0^{-1} \begin{bmatrix} x_0'' \\ y_0'' \\ 1 \\ d \end{bmatrix})$$

and therefore obtain pixel intensity information from all cameras in order to perform the matching.

During the stereo matching, each reference image point(x_0', y_0') and disparity value d has a set of associated reprojected pixel values that form a pixel value vector v. The value vector v is defined as

$$v(x_0', y_0', d) = \{ I_i(H(JW_i W_0^{-1} \begin{bmatrix} x_0'' \\ y_0'' \\ 1 \\ d \end{bmatrix})) \}, \; \forall i \in [1,...,n] \tag{3}$$

and will be used to derive the matching cost.

It is a well known fact that for a given camera geometry, each image point is restricted to move along a single line called the epipolar line. In the traditional approach to stereo matching, a single epipolar line A in one stereo image is matched with a corresponding epipolar line B in the other sterco image. The established matching between the two lines is a path in the grid of all possible matches (a,b), as shown on the left of Figure 1. The allowed starting and ending positions of the path are shown as thick black lines. By assuming that point ordering is maintained along epipolar lines, it is possible to solve this path problem very efficiently via dynamic programming.

With reference now to Figure 1 and in particular to the right side of that Figure, the matching grid is transformed into an equivalent formulation. In this case, each potential match has the form (a, d) where a is the position along the epipolar line and d the associated disparity. The coordinates in image B corresponding to the match (a,d) are easy to compute from Eq. 1 (i.e., from the relation between x_0' and $x_0' + bd$), while the cost function is directly obtained from Eq. 2.

An extension to matching a single pair of epipolar lines at a time would be to extend it to whole image at once, as depicted in Figure 2, by matching all pairs of epipolar lines simultaneously. Every minimum-cost path defining the matching of one pair of epipolar lines is now assembled into a single minimum-cost surface. This surface contains all the disparity information of the reference image. The goal of this construction is to take advantage of one very important property of disparity fields, local coherence, suggesting that disparities tend to be locally very similar. This property was exploited indirectly by using the ordering constraint (i.e., the assumption that point ordering is maintained, described above), along epipolar lines because the ordering constraint can be used under assumption of local coherence. However, local coherence occurs in all directions and thus across epipolar lines. By putting all the epipolar lines together and solving globally for a disparity surface, it becomes possible to take full advantage of local coherence and improve the resulting depth map.

Now that each potential match (a, l, d) in Figure 2, l denoting the number of a pair of epipolar lines, is still four-connected as in Figure 1. To take full advantage of local coherence, they should be six-connected to relate each individual epipolar line. Unfortunately, doing this makes dynamic programming unusable since there is no strict order for building the solution surface, i.e., since the surface shape is not restricted. (In the dynamical programming, in contrast, the solution path monotonically increases under assumption of the ordering constraint.)

Advantageously, the present method solves globally for the disparity surface by adding a source and a sink to the formulation of Figure 2, and treating the solution as a flow problem as depicted in Figure 3. Consider the graph G = (V, E) forming a 3D mesh as depicted in Figure 3. The vertex set V is defined as

$$V = V^* \cup \{s,t\}$$

where s is the source, t is the sink, and V^* is the 3D mesh

$$V^* = \{(x',y',d) : x' \in [0...x_{max}'], y' \in [0...y_{max}'], d \in [0...d_{max}'] \}$$

where $(x_{max}' + 1, y_{max}' + 1)$ is the image size and $d_{max}' + 1$ is the depth resolution. Internally the mesh is six-connected and the source s connects to the front plane while the back plane is connected to the sink t. The result for edges E is

$$E = \begin{cases} (u,v) \in V^* \times V^*: & \|u - v\| = 1 \\ (s,(x',y',0)): & x' \in [0 \ldots x'_{max}], \ y' \in [0 \ldots y'_{max}] \\ ((x',y',d_{max}),t): & x' \in [0 \ldots x'_{max}], \ y' \in [0 \ldots y'_{max}] \end{cases}$$

where u and v are vertices in the 3D mesh, equation $\|u-v\|=1$ represents an edge (u,v) connecting neighboring vertices, (s,(x',y'0) represents an edge connecting source s and a point (x',y',0) in the 3D mesh, and ((x',y',d$_{max}$),t) represents an edge connecting a point (x',y',d$_{max}$) in the 3D mesh and sink t.

Being six-connected instead of four-connected, each vertex of the new problem is not only connected to its neighbors along the epipolar line, but also across adjacent epipolar lines (see Figure 3). Since dynamic programming is useless in this situation, it is possible instead to compute the maximum-flow between the source and sink. The minimum-cut associated with the maximum flow will give the disparity surface directly.

In this context, the edge capacity c(u,v) represents the matching cost, except for edges adjacent to the source or sink, which have infinite capacities. The result is

$$c(u,v) = \begin{cases} 0 & if \ (u,v) \notin E \\ \infty & if \ u = s \ or \ v = t \\ reg(u,n) & if \ (u - v) = (0,0,\Delta_d) \\ occ(u,v) & if \ (u - v) = (\Delta_x, \Delta_y, 0) \end{cases}$$

where $(0,0,\Delta_d)$ represents a vector directing in the direction of the d-axis, $(\Delta_x,\Delta_y,0)$ represents a vector in the xy-plane, reg(u,v) is a cost characterized by the absence of occlusion (i.e. regular) while occ(u,v) is an occlusion cost. In Figure 3, the edges linking (x', y', d) to four black vertices are occlusion costs while links to gray vertices are regular costs. Define reg(u,v) from Equation 2 as

$$reg(u,v) = \frac{cost(u) + cost(v)}{2}$$

Higher occlusion costs increase the surface smoothness of the surface, and, inversely, lower occlusion costs facilitate depth discontinuities. In practice, set occ(u,v) to a natural value

$$occ(u,v) = reg(u,v)$$

or to a small constant larger than 0. Extreme values of the occlusion cost have intuitive consequences. When the occlusion cost is set to ∞, the resulting disparity surface is flat and featurs a single disparity value for the whole image. Setting the occlusion cost to 0, each pixel of the image is independently given a disparity, therefore achieving maximal discontinuity in the disparity surface.

It is well known that once the maximum flow is found, there is a cut C that separates the source and sink in such a way that the capacity of the cut, i.e., the sum of capacities of the edges, is minimized. The cut is therefore the optimal way to separate the source and the sink for the particular cost function. Since the source is connected to the closest points while the sink is connected to the deepest points, the cut effectively separates the view volume into a foreground and background and yields the depth map of the scene. The application of the maximum-flow and minimum-cut to solve the stereo correspondence problem has never been considered in the prior art. The minimum cut is also guaranteed to provide a depth estimate for each image point, as demonstrated by Property 1.

Property 1 (cut as a depth map)

Consider a cut C associated with some flow in the graph G = (V,E). For all (x,y), there exists at least one d such that the edge (x,y,d) - (x,y,d+1) is part of C. Proof. For any (x,y), there is a path s~t in G such that

$$s \rightarrow (x,y,0) \rightarrow (x,y,1) \ \tilde{} \ (x,y,d_{max}) \rightarrow t$$

therefore containing the set of edges

$$\left\{ \begin{array}{l} s \rightarrow (x,y,0) \\ (x,y,d) \rightarrow (x,y,d+1) \quad d \in [0,d_{max}\text{-}1] \\ (x,y,d_{max}) \rightarrow t \end{array} \right\}$$

where u → v stands for a directed edge directed from vertex u to vertex v. Any cut of G must break this path and thus contain at least one edge of the form (x,y,d)-(x,y,d+1) since the edges s→(x,y,0) and (x,y,d$_{max}$)→t have infinite capacities. (As a result of the infinite capacities of the edges associated with the source and the sink, a solution surface would never cut through those edges because the surface that cuts those edges would not be of a minimum capacity.)

According to property 1, a depth map can be constructed from the minimum-cut C of graph G as follows. For each point (x,y), the disparity is the largest d such that the edge (x.y,d)-(x,y,d+1) belongs to C. This results in the desired global disparity surface.

An important distinction has to be made between the stereo matching problems depicted in Figures 2 and 3. In the case of Figure 2, the epipolar lines are simply stacked up one after the other. While this might work for two-images stereo, it does not extend well to the case of multipl-image stereo image since the epipolar lines are specific to a single pair of cameras and arbitrary camera geometries will yield arbitrary set of epipolar lines.

To alleviate this problem, the ordering constraint is removed altogether. As mentioned earlier, the local coherence property approximates that constraint quite well. In the new formulation, it is possible to select any set of lines in the image to be stacked together. The obvious choice is to take the set of horizontal lines since this is the natural image ordering. This is why it is possible to refer to a point in Figure 3 by its image coordinates (x',y') instead of the epipolar line index I and a position a in Figure 2.

The epipolar geometry is now only indirectly used in computing the matching cost for points with given disparity values but does not contribute any actual constraint to the matching process.

A standard preflow-push algorithm, as described by T.H. Cormen, C.E. Leiserson, and R.L. Rivest, "Introduction to Algorithms". McGraw-Hill, New York, 1990, was used to solve the maximum-flow problem. The number of nodes in the graph is equal to the number of image pixels multiplied by the depth resolution. For an image of size n pixels, i.e., of dimension approximately $\sqrt{n} \times \sqrt{n}$, and a depth resolution of d steps, V = nd . Since the graph to solve is a three-dimensional mesh where each node is six-connected, the number of edges is E ≈ 6V. This means that in practice, algorithms solving maximum flow in $O(VE^2)$, $O(V^2E)$, or $O(V^3)$ are all equivalent on this problem. Currently, only the preflow-push lift-to-front algorithm (which is $O(V^3)$) was tested but other preflow-push algorithms should perform satisfactorily.

In the term of image dimensions, $O(V^3)$ translates into $O(n^3d^3)$. The standard dynamic programming approach on separate epipolar lines requires a total running time of θ(nd). This might seem much better than $O(n^3d^3)$. However, the topology of the graph, the position of source and sink, and the nature of edge capacities all tend to make the problem easier to solve. Figures 4 and 5 show the typical performance of the present invention as a function of image size n (in pixels) and depth resolution d, the x-axis and the y-axis being log(n) and log(time),respectively. In the figures, "Discharge" and "Lift" refer to well-known basic operations in the preflow-push maximum flow algorithm, and "Run time" stand for the actual computer time to run the preflow-push algorithm to solve a stereo vision problem. The average running time is $O(n \, d^{1.4})$, which is linear with respect to image size n (in pixels) and compares favorably with the dynamic programming approach.

To demonstrate the symmetry in the disparity map achieved by the maximum-flow method, it was applied to a conventional random-dot stereogram as shown in Figure 6 with disparities set at O, 4 and 8 pixels.

In Fig. 6, the center of the image is shifted 8 pixels in the left and right image, the outer portion is shifted 0 pixels between the left and right images and intermediate portions are shifted 4 pixels between the left and right images. If the left and the right image are fused, then the result would be three floating rectangular planes. The images shown in Fig. 6 are a standard pair of images for testing people for what is referred to as stereo blindness.

The resulting disparity maps, shown in Figure 7A and 7B for maximum-flow and standard stereo mapping methods respectively, differ primarily around depth discontinuities. The maximum-flow method features similar boundaries in all directions while the standard stereo yields a very different boundary shape, due to the fact that solutions are computed horizontally and no information is shared vertically.

Figures 9A - 9D are a set of disparity maps for shrub images shown in Figure 8. The maximum flow maps (the minimum cut maps) are shown in Figure 9A and Figure 9C. Standard stereo maps are shown in Figure 9B and Figure 9D. Figure 9A and Figure 9B are 32 disparity step resolution and Figure 9C and Figure 9D are 128 disparity step resolution.

It should be noted that the maximum-flow maps (Figures 9A and 9C) tend to extract sharp and precise depth discontinuities while the standard stereo maps (Figures 9B and 9D) produce many artifacts along vertical depth discontinuities. The level of depth resolutions show different levels of smoothness. Even at high smoothness levels, maximum flow does not produce spurious horizontal lines across the gap between the two larger shrubs. The brand image that

appears in Fig. 8 is the non-occluded match.

In the simplest case, the occlusion cost is the regular non-occluded cost. It is set to a very high value in order to discourage occlusion and favor surface smoothness. In the opposite extreme, this cost is set to the smallest possible value allowing a non trivial solution, i.e. the constant 1. Figure 10A, Figure 10B and Figure 10C illustrate how changing the occlusion cost can affect the disparity maps obtained. As expected, lowering the capacities favors depth discontinuities and therefore creates sharper object edges, at the expense of smoothness of surfaces. It is observed, however, that depth discontinuities tend to be sharper than other methods, for a given level of smoothness. This result differs strongly from most other methods where high level of smoothness correspond to blurred or missing depth discontinuities.

While the above description generally refers to plural cameras and a single scene, in fact, the invention applies to an arbitrary sequence of images of a common scene. The invention also applies to a sequence taken by one or more cameras at a single instance or over a period time. For example, two or more synchronized still cameras obtaining an image at the same instance, or a single camera taking images over a time period, e.g. one or more still cameras obtaining images while the camera(s) is/are in motion or a video camera which is in motion.

In conclusion, it is described how to use a vector of pixel values to derive a meaningful matching cost. If it is assumed that surfaces are lambertian (i.e. with a surface reflectance independent of light direction), then the component intensity values of vector $v(x', y', d)$ should be the same if (x', y', d) is on the surface of an object, because the view angle or the illumination angle has no effect because of the assumption of a lambertian surface. The matching cost $c(x', y', d)$ is then defined as the variance of vector $v(x', y', d)$ of n elements, that is

$$c(c',y',d) = \frac{1}{n}\Sigma(v-\bar{v})^2$$

With reference now to Figure 13, there is shown a flow chart which depicts the overall method of the present invention. Specifically, and with reference to that Figure, at least two cameras provide a sequence of input images to the method (steps 1301 and 1302). A graph (G) is created (step 1304) composed of a step of establishing vertices (V) (step 1306) and a step of establishing edges (E) linking these vertices (step 1308) and a step of establishing capacities of the edges or establishing a capacity function (c(u,v)) (step 1310).

Then a maximum-flow is determined in the graph (G) using standard flow method (preflow-push) (step 1320). The minimum-cut is extracted from the determined maximum-flow (step 1330). The disparity surface (depth map) is computed from the extracted minimum-cut (step 1340). The final disparity surface is derived from the computed disparity surface in step 1340 (step 1350). The result is a disparity surface which is the stereo correspondence sought.

While there has been described and illustrated a method for performing stereoscopic correspondence between two or more arbitrary views of a scene, it will be understood by those skilled in the art that variations and modifications are possible without deviating from the spirit and broad teachings of the present invention which shall be limited solely by the scope of the claims appended hereto.

Claims

1. A method of obtaining stereo correspondence between multiple views of a scene from a sequence of images thereof;
characterized by the steps of:

composing 3-dimensional mesh V* associated with an x'-, a y'- and a disparity-axis with vertices thereof corresponding to a set of x', y' coordinates and disparities of the images;
composing a set of vertices V of source, sink and the set of vertices of said 3-dimensional mesh V*;
creating a set of edges E by linking vertices in said set of vertices V*, linking said source with the front plane of said 3-dimensional mesh as viewed in the direction of said d-axis, and linking said sink with the back plane of said 3-dimensional mesh as viewed in the direction of said d-axis;
assigning capacity functions that represent matching costs to edges in said 3-dimensional mesh V*;
assigning capacities to edges linking the source with said front plane of said 3-dimensional mesh, where the capacity of any edge linking the source with said front plane of said 3-dimensional mesh is sufficiently large so that the capacity of any cut that includes at least one edge from said source to the front plane of said 3-dimensional mesh V* will be larger than the capacity of any cut that passes through said 3-dimensional mesh to separate said source and sink and that includes neither an edge from said source to the front plane nor an edge linking said sink and said back plane;
assigning capacities to edges linking said sink with said back plane of said 3-dimensional mesh, where the capacity of any edge linking said sink with said back plane of said 3-dimensional mesh is sufficiently large so

that the capacity of any cut that includes at least one edge linking said back plane of said 3-dimensional mesh V* with said sink will be larger than the capacity of any cut that passes through said 3-dimensional mesh to separate said source and sink and that includes neither an edge linking said sink and said back plane nor an edge from said source to the front plane;

finding the maximum flow in the graph G(V,E) created through the above-described procedures;

extracting the minimum cut from said maximum flow in said graph G(V,E); and

computing a disparity surface from said minimum cut.

2. A method as set forth in Claim 1, wherein said three-dimensional mesh V* is six-connected.

3. A method as set forth in Claim 1, wherein said set of vertices V is

$$V = V^* \cup \{s, t\},$$

and

$$V^* = \{(x', y', d): x' \in [0...x'_{max}], \ y' \in [0...y'_{max}], \ d \in [0...d_{max}]\},$$

$$E = \begin{cases} (u,v) \in V^* \times V^*: & \|u - v\| = 1 \\ s,(x',y',0)): & x' \in [0...x'_{max}], \ y' \in [0...y'_{max}] \\ s,(x',y',0)): & x' \in [0...x'_{max}], \ y' \in [0...y'_{max}] \end{cases}$$

and said capacity function c(u,v) is:

$$c(u,v) = \begin{cases} 0 & if \ (u,v) \notin E \\ \infty & if \ u = s \ or \ v = t \\ reg(u,n) & if \ (u - v) = (0,0,\Delta_d) \\ occ(u,v) & if \ (u - v) = (\Delta_x, \Delta_y, 0) \end{cases}$$

where s and t are said source and said sink, respectively, $(x'_{max} + 1, y'_{max} + 1)$ is the image dimension, $d_{max} + 1$ is the depth resolution, $\|u-v\|$ is a length of the edge linking neighboring vertices, $reg(u,v)$ is a capacity of edge (u,v) representing a matching cost characterized by the absence of occlusion, $occ(u,v)$ is a capacity of edge (u,v) representing an occlusion cost, Δ_d is a length of an edge (u,v) in the direction of the disparity axis, and Δ_x and Δ_y are the components of an edge in the x'y'-plane.

4. A method as set forth in Claim 1, where said sequence of images is obtained by a plurality of cameras.

5. A method as set forth in claim 1, where said sequence of images is obtained by two or more cameras at a single instance of time.

6. A method as set forth in claim 1, where said sequence of images is obtained by a one or more cameras over a period of time.

7. A method as set forth in claim 1, where said sequence of images is obtained by a one or more cameras in motion.

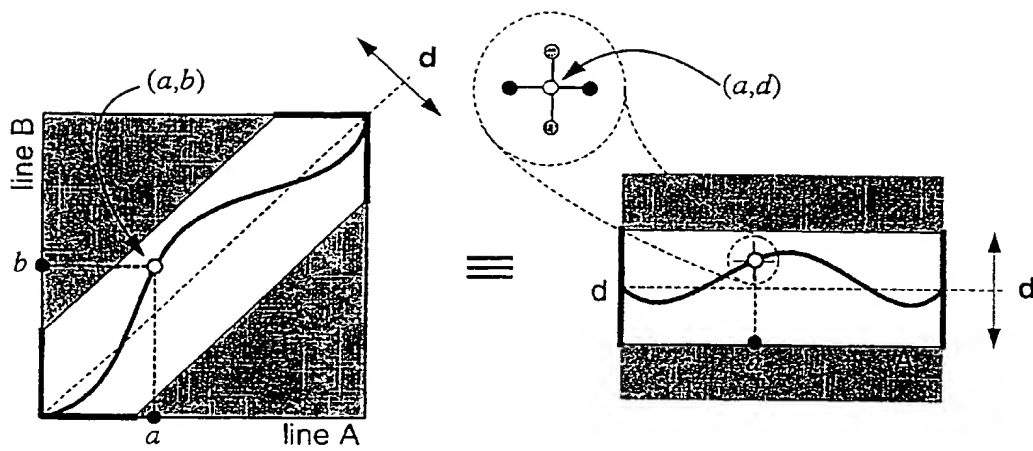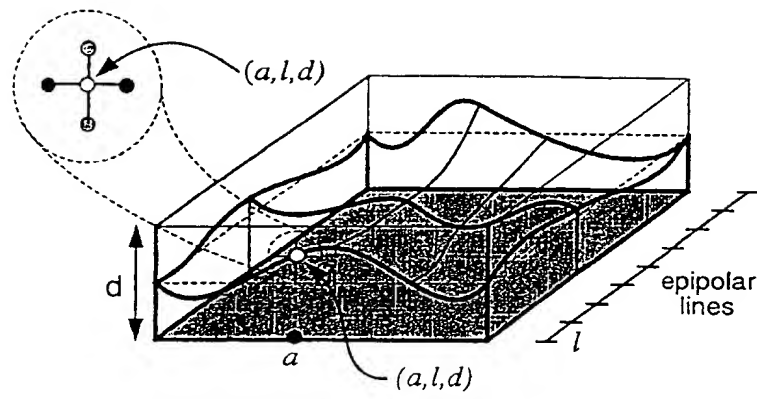8. A method as set forth in claim 1, where said sequence of images is obtained by one or more video cameras in motion.
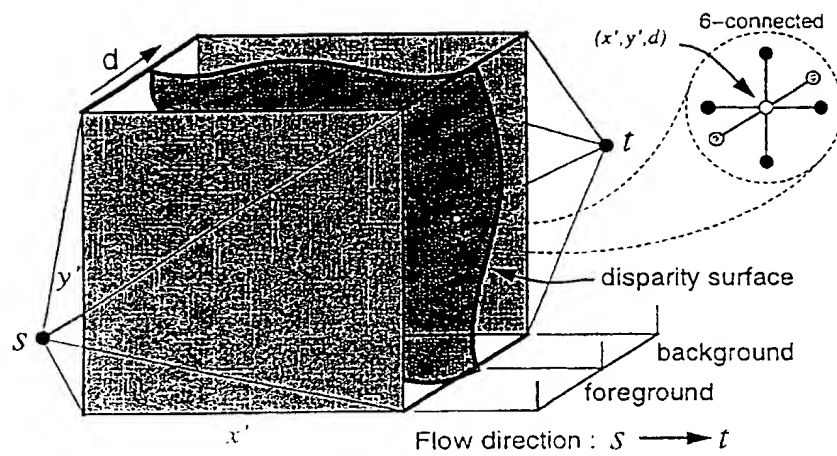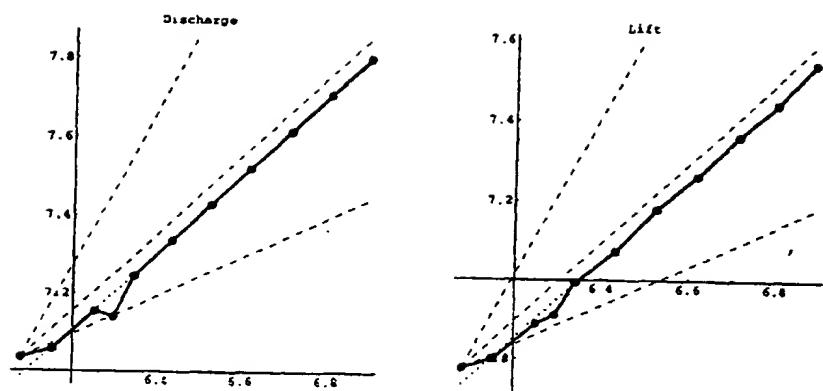
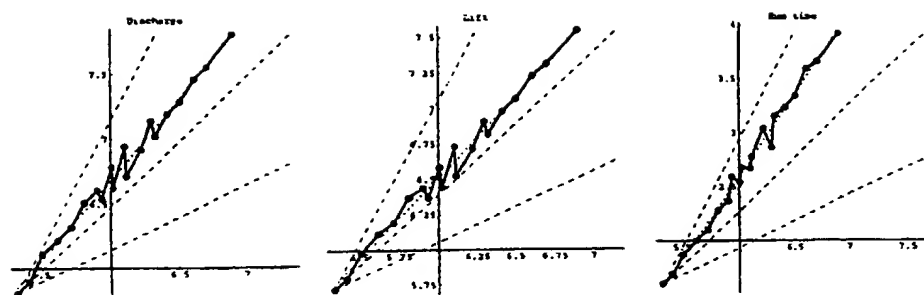Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

right   left

Figure 6

256 x 256 (32 disparity steps)

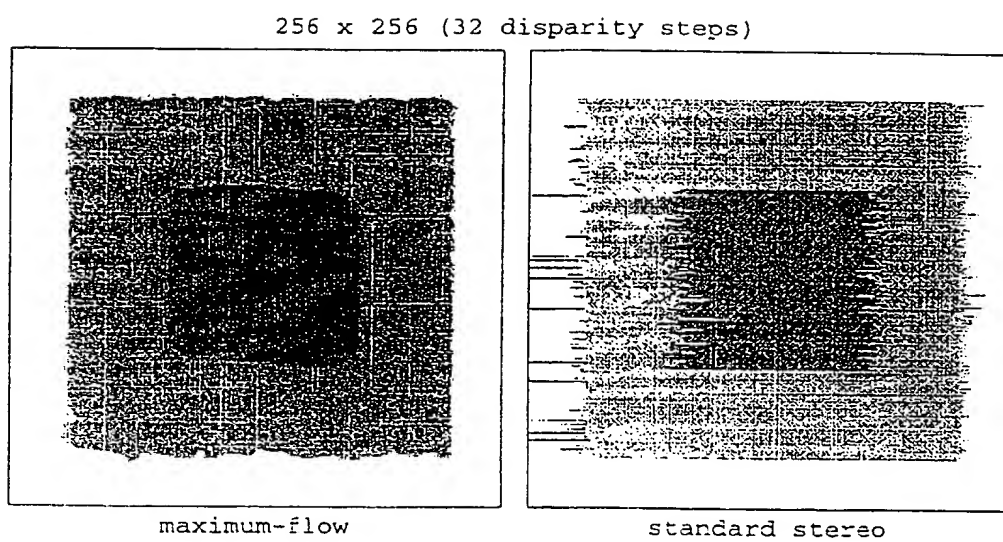maximum-flow   standard stereo

Figure 7
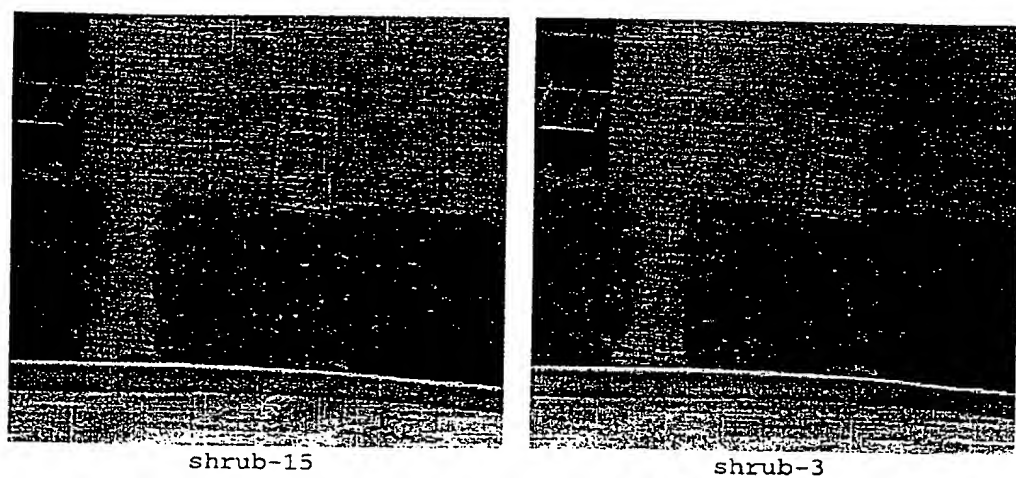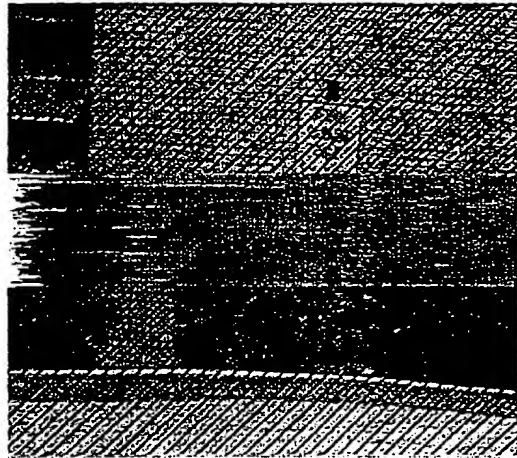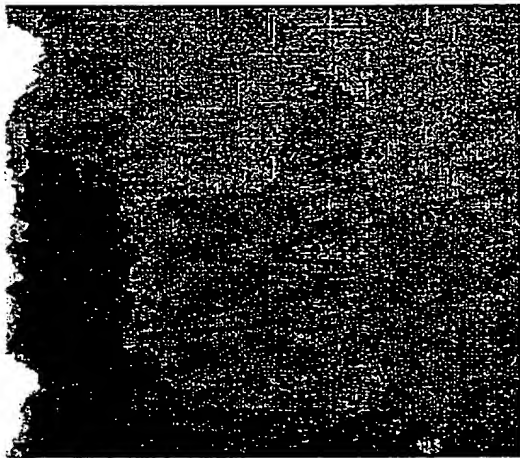
shrub-15   shrub-3

Figure 8

256 x 240 (32 disparity steps)

256 x 240 (128 disparity steps)
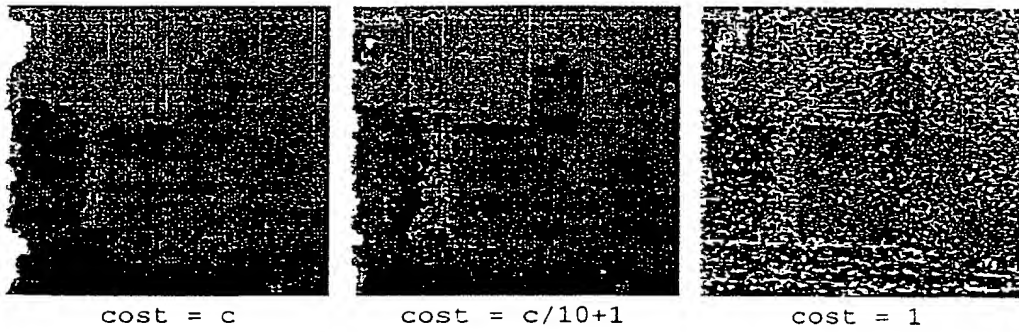
maximum-flow   standard stereo

Figure 9

cost = c   cost = c/10+1   cost = 1

Figure 10

Figure 11

A)

I_1 I_2

C_0 C_1 C_2

B)

I_1 I_2

C_1 C_0 C_2

Figure 12

1301   1302

Camera 1   Camera 2

1304

Create Graph: $G = (V, E)$ where

$V = V^* \cup \{s, t\}$ 1306

$V^* = \{(x', y', d) : x' \in [0 \ldots x'_{max}], \ y' \in [0 \ldots y'_{max}], \ d \in [0 \ldots d_{max}]\}$

$E = \left\{ \begin{array}{lll} (u, v) \in V^* \times V^* & : & \|u - v\| = 1 \\ (s, (x', y', 0)) & : & x' \in [0 \ldots x'_{max}], \ y' \in [0 \ldots y'_{max}] \\ ((x', y', d_{max}), t) & : & x' \in [0 \ldots x'_{max}], \ y' \in [0 \ldots y'_{max}] \end{array} \right.$ 1308

$c(u, v) = \left\{ \begin{array}{ll} 0 & \text{if } (u, v) \notin E \\ \infty & \text{if } u = s \text{ or } v = t \\ \text{reg}(u, v) & \text{if } (u - v) = (0, 0, \Delta_d) \\ \text{occ}(u, v) & \text{if } (u - v) = (\Delta_x, \Delta_y, 0) \end{array} \right.$ 1310

Find Maximum–Flow in graph G
using standard flow method (preflow–push)   1320

Extract Minimum–Cut from obtained Maximum–Flow   1330

Compute Disparity Surface (depth map) from Minimum–Cut   1340

Disparity Surface   1350

Figure 13